Personalized Handling of Semantic Data with MIG

Mariano Rico, David Camacho
Computer Science Department
Universidad Autónoma de Madrid, Spain
{mariano.rico, david.camacho}@uam.es

Óscar Corcho
Ontology Engineering Group
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid, Spain
ocorcho@fi.upm.es

Abstract

This paper shows a semantically-enabled web application named MIG used to create user profiles which enhances users accessibility by allowing the creation of an user interface adapted to the user needs, the device used, and its preferences. This approach exploits the Semantic Web technologies and the infrastructure and applications created in previous work.

1. Introduction

According to d'Aquin et al [1], in 2007 there were around 23,000 ontologies available on the Internet, and semantic data have grown exponentially for the last ten years [2], especially after the Linked Open Data initiative [1]. And the maturity of semantic technologies is demonstrated by its use in a wide number of applications nowadays [3].

However, despite this wealth of information, the Semantic Web envisioned by its authors is still far away. In their envision [4], semantic agents specialized in human-computer interaction are able to act as our majordomes in the Web. These agents should create an user interface adapted to the user needs, specifically its interaction device (e.g. PC, handheld or TV), its characteristics (e.g. color blindness or reduced visual sharpness), or its preferences (e.g aesthetic or corporative image). This may seem an unnecessary requirement, but semantic agents must be able to solve this problem because in the vision of the Semantic Web, the semantic agents specialized in interacting with humans are responsible for creating a suitable user interface.

Figure 1 shows the change in the way users and companies interact. Part (a) of this figure shows the current situation, in which users interact directly with companies by means of web applications. Although there exist web hubs, such as those for finding flight tickets, the most common situation is a one-on-one relation between a user and a given company through the company web site. The Semantic Web promises semantic agents able to aggregate semantic data from different companies and offer to the users a unified user interface, as shown in part (b). The big deal for these companies is that creating a web site is not mandatory, and they have to focus only on providing semantic data.

1. See http://linkeddata.org

The approach presented in this paper is aimed at bridging this gap by using a semantically-enabled application named MIG in conjunction with VPOET [5], as shown in part (b) of figure 1. MIG allow users create and store user profiles, with data about its interactive needs, its interaction device, and its aesthetic preferences. VPOET stores web templates created by web designers. The hypothetical semantic agent shown in part (b) of figure 1 could exploit the information provided by VPOET and MIG to create a personalized web interface.

The next section shows briefly Fortunata and VPOET. Section 3 shows how VPOET can exploit MIG profiles to provide the "most adequate"template for a given user profile. Section 4 presents related work, and section 5 presents conclusions and future work.

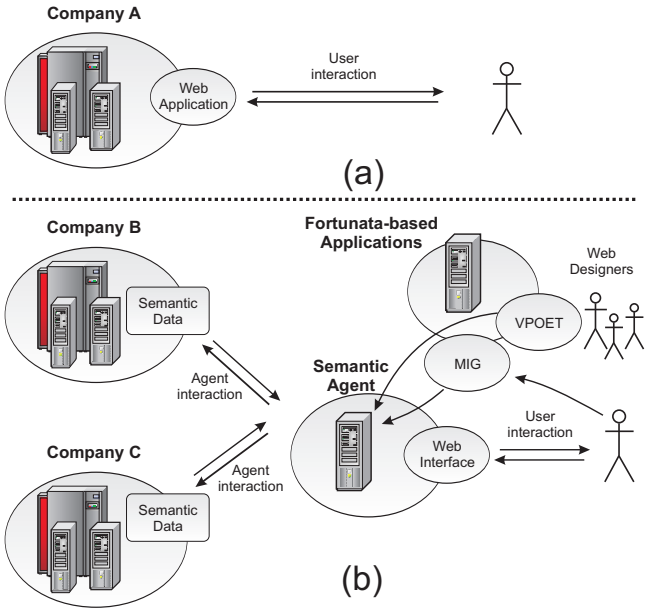

Fig. 1. Changes in the relationships between users and companies due to the Semantic Web paradigm. (a) Traditional way. (b) Semantic Web envision and our approach.

2. Fortunata and VPOET

This section summarizes Fortunata [6], the underlying infrastructure used by MIG, as well as other Fortunata-based

application named VPOET which exploits the semantic user profile stored by MIG.

2.1. Fortunata to Create Semantically-Enabled Web Applications

Fortunata [2] is a Java library built on top of the JSPWiki wiki engine, whose main features are its support for the management of forms and its extensibility capabilities by means of plugins.

Fortunata simplifies the creation of semantically-enabled web applications by delegating to the underlying wiki engine the client-side presentation and server-side publication of semantic data. The creation of pages is done with a wiki-based syntax, which has predefined constructs to create links, sortable tables, tables of contents, etc. The publication of semantic data is done automatically by the system, which also includes an easy-to-use mechanism to add links to these data from any wiki page.

A Fortunata-based application consists in a set of wiki pages that contain regular wiki code intertwined with calls to F-plugins. For instance, VPOET is a Fortunata-based application that consists in four interrelated wiki pages and seven F-plugins.

The main assumptions that motivated our work in Fortunata were that we can clearly separate the roles involved in the creation of semantically-enabled web applications and that the skills required for this can be drastically reduced if we provide adequate tools for each of these roles.

2.2. VPOET to Handle Semantic Data in Web Applications

VPOET is a Fortunata-based web application oriented to enable client-side web designers, also known as "template providers", to create web templates for a set of ontology components. These templates can be used to visualize semantic data (output templates) or to request it from users (input templates). For example, let's imagine that we want to create output and input templates for the concept `Person` in the `FOAF` ontology. These templates can be used to render any data source containing instances (individuals) of this class (or any subclass if there are no more specialized templates for them), and to present a form to request data that will be converted to an instance of `Person`, respectively.

VPOET is focused on Web designers, who should be able to author attractive designs capable of handling semantic data. Hence, VPOET only requires basic skills in client-side technologies (e.g., HTML, Javascript). The most difficult task to be performed by such developers is to embed some semantic data management macros in the client-side web code (HTML, CSS, or Javascript) generated by the web designer favorite authoring tool (eg. Dreamweaver). Hence there is little training needed to start creating templates (a 30 min. online tutorial [3] is enough, as showed in our evaluation).

From the point of view of end users who browse through the visualization of semantic data sources generated by output templates or who have to introduce semantic data with input templates, a VPOET-enabled application is like any other web application, with information shown in tables or any other HTML element, and usual HTML forms with text fields, radio buttons, etc.

VPOET has two faces, on the one hand it is a web application oriented to web designers ranging from amateur users to professional ones. On the other hand, it is a semantic data source fed by the templates created by a community of web designers sharing and reusing templates.

This source can be exploited easily by common web developers, in any programming language, by means of HTTP messages (GET and POST) like "render the semantic data at URL Z by using the output template X created by designer Y", codified as a HTTP GET message by means of the following URL:

```
http://URL-to-servlet/VPoetRequestServlet?↵
action=renderOutput&↵
designID=X&↵
provider=Y&↵
source=Z
```

Note: The symbol ↵ means that the URL has been splitted for readability.

An additional argument `indvID` specify a given individual in the source. In this case, only the individual is rendered. The full syntax of these macros and HTTP messages can be found in the aforementioned tutorial.

A Google Gadget named GG-VPOET [4] exploits VPOET templates by means of the aforementioned HTTP messages. By using this gadget, any end user can render a semantic data source or provide a web interface to create semantic data. GG-VPOET, as any other Google Gadget, can be inserted into a regular web page or in Google products such as iGoogle, Google Desktop or Google Pages.

2.3. Extending VPOET to Support Personalization

Following the previous example, if the parameters `designID` and `provider` (that specify uniquely a template) are missing, and the parameter `object` (e.g. `FOAF.Person`) is specified, VPOET retrieves the templates designed for that object and uses anyone of them to render the given semantic data source. But if the parameter `userProfile` is specified, VPOET should return the most adequate template for that user profile. This parameter is a URL pointing to a MIG user profile. The following HTTP message shows an example:

```
http://URL-to-servlet/VPoetRequestServlet?↵
action=renderOutput&↵
source=Z&↵
```

2. See http://ishtar.ii.uam.es/fortunata

3. See http://ishtar.ii.uam.es/fortunata/Wiki.jsp?page=VPOETTutorial

4. Available at the Google Gadgets Directory (http://www.google.com/ig/directory?type=gadgets)

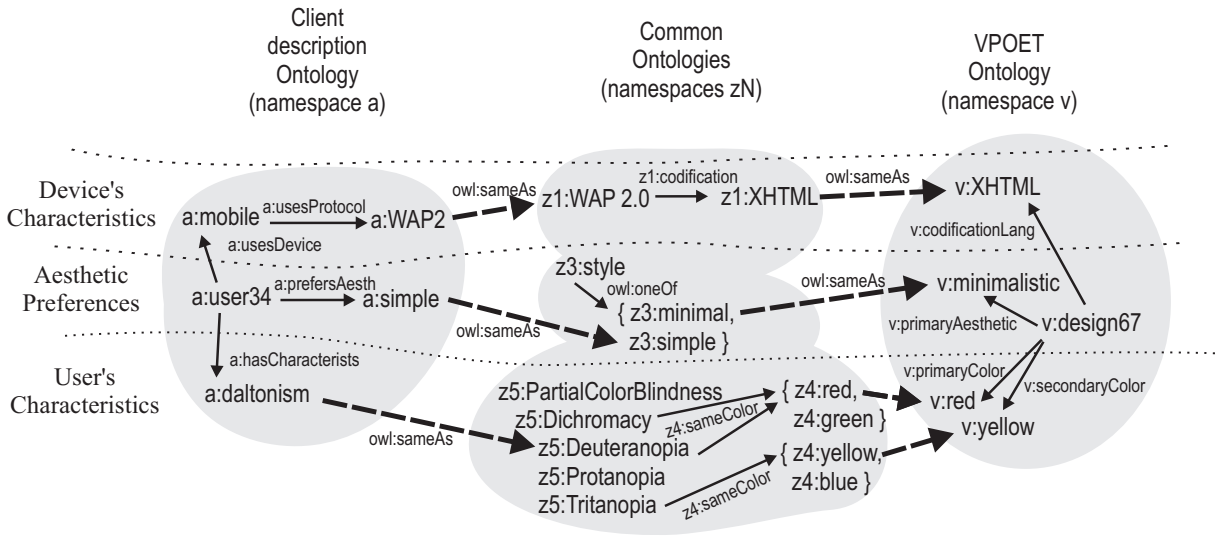

Fig. 2. Integrating semantic data from VPOET and MIG.

object=foaf.Person&↵
userProfile=http://URL-to-MIG/mig.data.owl#profile89
Note: The symbol **#** must be codified as **%23** to be parsed correctly by VPOET.

3. Creating a User Profile with MIG

MIG [5] stands for "Me InteractinG". It is a Fortunata-based web application oriented to common users interested in specifying their profile. This profile comprise details about vision impairment, device used and aesthetic preferences, as shown in figure 3. As any other Fortunata-based application, MIG stores the information provided by its users as semantic data easily reachable at a well known URL.

An example of matching VPOET templates and a given MIG user profile is depicted in figure 2. Each ontology, identified by a namespace, is shown as a cloud. The elements of the ontology, and their individuals, are shown inside its cloud. The left part of this figure shows the ontology describing the user profile, characterised by namespace a. In this example, the user identified as *a:user34* has the following profile: (1) uses a WAP2 mobile phone as interaction device, (2) prefers simple aesthetics and (3) he/she is daltonic (colour-blindness associated to red-green colours).

In centre part of figure 2, public well-known ontologies are shown. Ontology z_1 indicates that the protocol WAP2.0 is codified as XHTML. For ontology z_3,"minimal"and "simple"are different kinds of styles but semantically close. Ontology z_5 has a visual-impairments hierarchy.

The right part of figure 2 shows the VPOET ontology, with namespace v. In this ontology, the template identified as *v:design67* is codified using the XHTML language, its primary aesthetic is minimalistic, and it has red and yellow as primary and secondary colours.

5. See http://ishtar.ii.uam.es/fortunata/Wiki.jsp?page=MIG

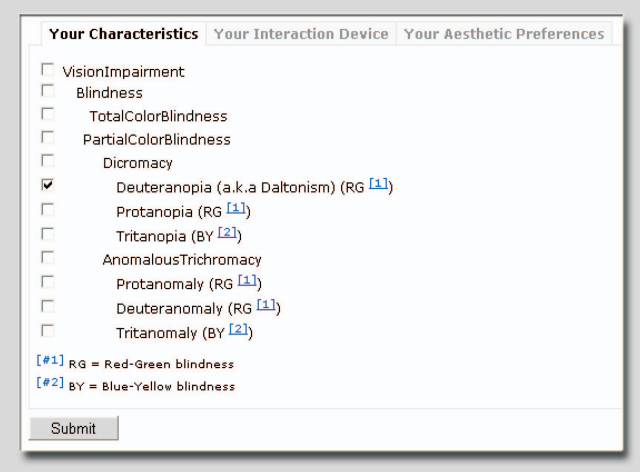

Fig. 3. User characteristics in MIG

With this semantic information is impossible to find that *v:design67* is a valid template for *a:user34*. An additional semantic data source is required in order to link elements belonging to different ontologies. These links use to be "sameAs"(technically there are three types: owl:sameAs, owl:equivalentClass and owl:equivalentProperty to distinguish individuals, classes, and properties/relations respectively) relations, shown as discontinuous bold arrows in figure 2. Joining all this semantic information, a semantic query (e.g., by using SPARQL language) based in the user profile, like this one: "select a template with these characteristics: (1) codified in XTHML, (2) with minimalism as chief aesthetic, and (3) with primary colors avoiding red and green tones for text and background"would return a personalized template. For this example, the result of this query would be the design *v:design67*.

3.1. Public Ontologies Evaluation

The Linked Data initiative recommends reusing ontologies. Although we have used *ad hoc* ontologies for VPOET and MIG, we have reviewed some public ontologies and some parts have been used by MIG.

The MIG visual impairment part has been inspired by the works of Karim & Tjoa [7], the Digital Item Adaptation (DIA) part of MPEG-21, and the Desease Ontology [6]. It is remarkable that most industrial standards, such as MPEG-7 and MPEG-21, provide detailed descriptions of the user and the user's environment, but using XML Schema to define their models. Although some initiatives [8][9] have considered a semantic version of these standards, their results are not publicly available. In the conclusions section we point out a promising solution.

The MIG user device characterization comprises technical details such as display size, color bits per pixel, and browser type. Most modern browsers send the "User-Agent"(UA) string to the server in each HTTP message. MIG exploits this feature in order to detect the browser type, the Operating System, and the device's model when a device is used. MIG compares the UA string sent by the user's browser to WURFL [7], a UA's repository. This repository stores information about 9000 devices, with hundreds of possible capabilities (e.g. `is_wireless_device` or `resolution_width`). If the UA is found in the repository, most data concerning the terminal capabilities can be obtained from the repository. The current MIG implementation only considers display size and color bits per pixel.

The MIG user preferences is an arbitrary taxonomy with concepts such as *simple*, *baroque* or *minimalist*. The user profile provides an ordered list of preferred aesthetics.

3.2. Finding the Best Template

As pointed before, the model obtained from merging the semantic information from VPOET, MIG, common ontologies and linking elements, can be queried by means of semantic querying languages such as SPARQL. Although the results of a given query depend on the information stored in the model, as we saw in the matching example, and the same query can return 0 or many results depending on key linking elements, what happens when many results match the query?. The only way we have to constrain the results to reach "the best"template for a given user profile is adding more parameters to the SPARQL query.

The problem with using one SPARQL query is that it can return many results, with no sorting criteria, o none at all. The first case denotes a query too relaxed, and the second one a query too restrictive.

The solution adopted considers a set of SPARQL queries, ordered from less restrictive to more restrictive, i.e. with few parameters to more parameters, according to an importance criteria. For example, the first query can request matching display size, the second request matching display size and browser type, and so on. When the first query is fired, if it returns more than one template, the second one is fired, following this process until no results are found. The last query with results is considered the "best", and anyone of its resulting templates is used to render the given semantic data source.

6. See http://diseaseontology.sourceforge.net
7. See http://developer.openwave.com/dvl/tools_and_sdk/wurfl_and_wall

4. Related Work

The personalization topic can be reviewed in [10]. End user preferences applied to Semantic Web Services can be found in [11], where different aspects such as user current context, history (usage and context) or corporate data are considered in order to create user interfaces for cellular phones. End user preferences applied to the creation of ontologies is addressed in [12], where the concept of viewpoint is defined.

Adaptive interfaces [13], characterized by their explicit ability to adapt to the end user is a main topic in Human-Computer Interaction. An extension of the user model, named behavior-based, is used for personalized web browsing in [14], where the user profile contains information about browsing goal, interest, expertise and browsing behavior. However the interface is not personalized in the sense of adaptation to the user needs. The approach followed by SADIe [15] considers semantic annotation of CSS to facilitate the transcoding of a given annotated web page to the requirements of impaired users. Our approach do not considers transcoding but template selection based in the user needs.

Concerning technology, Digital Item Adaptation (DIA) [16] is Part-7 of the MPEG-21 standard. DIA bridges the mismatch between rich multimedia content and the usage environment. To this end, descriptions of contents and usage environments are provided in XML. These descriptions are modelled conforming to appropriate XML Schemas by considering four aspects: Terminal Capabilities, Characteristics from User, Network and Natural Environment. Accessibility aspects such as visual impairment are considered in great detail (e.g. "LowVisionSymptoms"is comprised of "LoosOfFineDetail", "LackOfContrast", "LightSensitivity", "NeedOfLight", "CenterVisionLoss", "PeripherialVisionLoss"), all of these with a numerical value to indicate the impairment degree. This specification has not been ontologized.

5. Conclusions and Future Work

Many aspects must be solved in order to achieve the semantic agents specialized in interacting with end users envisioned in the Semantic Web. This work presents an easy to develop and extend framework and web applications oriented to provide developers with a simple HTTP messages based mechanism to provide a web interface personalized to the user profile for handling semantic data.

We have merged the ontologies and data from VPOET and MIG, as well as the appropriated "sameAs"equivalences in order to get a unique model. This model is queried by means of SPARQL queries to obtain matching templates for a given user profile. The way to choose the "best"template is an open issue, in which Semantic Web Rules (SWRL [8]) or other non-semantic techniques such as classification algorithms, e.g. Clustering (KNN, K/X Means) or Support vector machines(SVM), can be applied.

Future work will deal with templates composition, template-template interaction, and many technical details such as support for client-side languages such as Action Script (to provide users with rich-interfaces based in Flash) or XHTML. Related works, such as the XML2RDF[9] tool, could help us to generate ontologies from XML Schemas (such as MPEG-7 and MPEG-21) in order to convert parts of the *ad hoc* ontology used into a more standards based ontology, following the Linked Data initiative.

Visit the evolution of this framework and its semantically-enabled applications at http://code.google.com/p/fortunata/.

Acknowledgment

This work has been partially funded by the Spanish Ministry of Science and Innovation under the projects HADA (TIN2007-64718), METEORIC (TIN2008-02081) and DEDICON (TIC-4425).

References

[1] M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta, "Characterizing Knowledge on the Semantic Web with Watson," in *Proc. EON2007, ISWC/ASWC*, vol. 329. CEUR Workshop Proceedings, 2007, pp. 1–10.

[2] T. Finin and L. Ding, "Search engines for semantic web knowledge," in *Proc. XTech 2006: Building Web 2.0*, 2006.

[3] M. Davis, "Semantic Wave 2008 Report: Industry Roadmap to web 3.0 & Multibillion Dollar Market Opportunities. Executive Summary." ProjectX10, Tech. Rep., 2008. See www.project10x.coms

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 28–37, 2001.

[5] M. Rico, D. Camacho, and Óscar Corcho, "VPOET: Using a Distributed Collaborative Platform for Semantic Web Applications,". Proc. IDC'2008, vol. 162. Springer, SCI Series, 2008, pp. 167–176.

[6] M. Rico, D. Camacho, and Óscar Corcho, "A Contribution-based Framework for the Creation of Semantically-enabled Web Applications," *submitted for publication to Journal of Information Sciences*, 2009.

[7] S. Karim and A. M. Tjoa, "Towards the use of ontologies for improving user interaction for people with special needs," *LNCS*, vol. 4061, pp. 77–84, 2006.

[8] J. Hunter, *Adding Multimedia to the Semantic Web: Building and Applying an MPEG-7 Ontology*, John Wiley & Sons Ltd, 2005.

[9] R. Troncy, W. Bailer, M. Hausenblas, P. Hofmair, and R. Schlatte, "Enabling Multimedia Metadata Interoperability by Defining Formal Semantics of MPEG-7 Profiles," in Proc. SAMT 200, LNCS vol. 4306, 2006, pp. 41–55.

[10] J. O. Blom and A. F. Monk, "Theory of Personalization of Appearance: Why Users Personalize their PCs and Mobile Phones," *Human-Computer Interaction*, vol. 18, no. 3, pp. 193–228, 2003.

[11] D. Khushraj and O. Lassila, "Ontological Approach to Generating Personalized User Interfaces for Web Services," *Proc. ISWC*, vol. 3729, pp. 916–927, 2005.

[12] R. Thomopoulos, "Expressing preferences in a viewpoint ontology," *Proc. COOPIS, DOA, and ODBASE, PT 2*, vol. 3761, pp. 1596–1604, 2005.

[13] M. Florins, "Graceful Degradation: a Method for Designing Multiplatform Graphical User Interfaces," Ph.D. dissertation, Université catholique de Louvain, 2006.

[14] M. Sah, W. Hall, and D. C. D. Roure, "Designing a Personalized Semantic Web Browser," in *Proc. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2008)*, LNCS vol. 5149, pp. 333-336, 2008.

[15] S. Harper and S. Bechhofer, "Sadie: Structural Semantics for Accessibility and Device Independence," *ACM Trans. Comput.-Hum. Interact.*, vol. 14, no. 2, p. 10, 2007.

[16] A. Vetro and C. Timmerer, "Digital Item Adaptation: Overview of Standardization and Research Activities," *IEEE Transactions On Multimedia*, vol. 7, no. 3, pp. 418–426, Jun. 2005.

8. See http://www.w3.org/Submission/SWRL/
9. See http://rhizomik.net/redefer